

Literatur zu Linux

- im Internet
- Teil der Installation: man, info
- O'Reilly Verlag:
 - Siever, Figgins & Weber:
Linux in a Nutshell 4th Edition
- Addison-Wesley:
 - Helmut Herold: Linux/Unix Grundlagen, Kommandos und Konzepte 5. Auflage
- noch viele weitere Bücher und Online-Dokumente

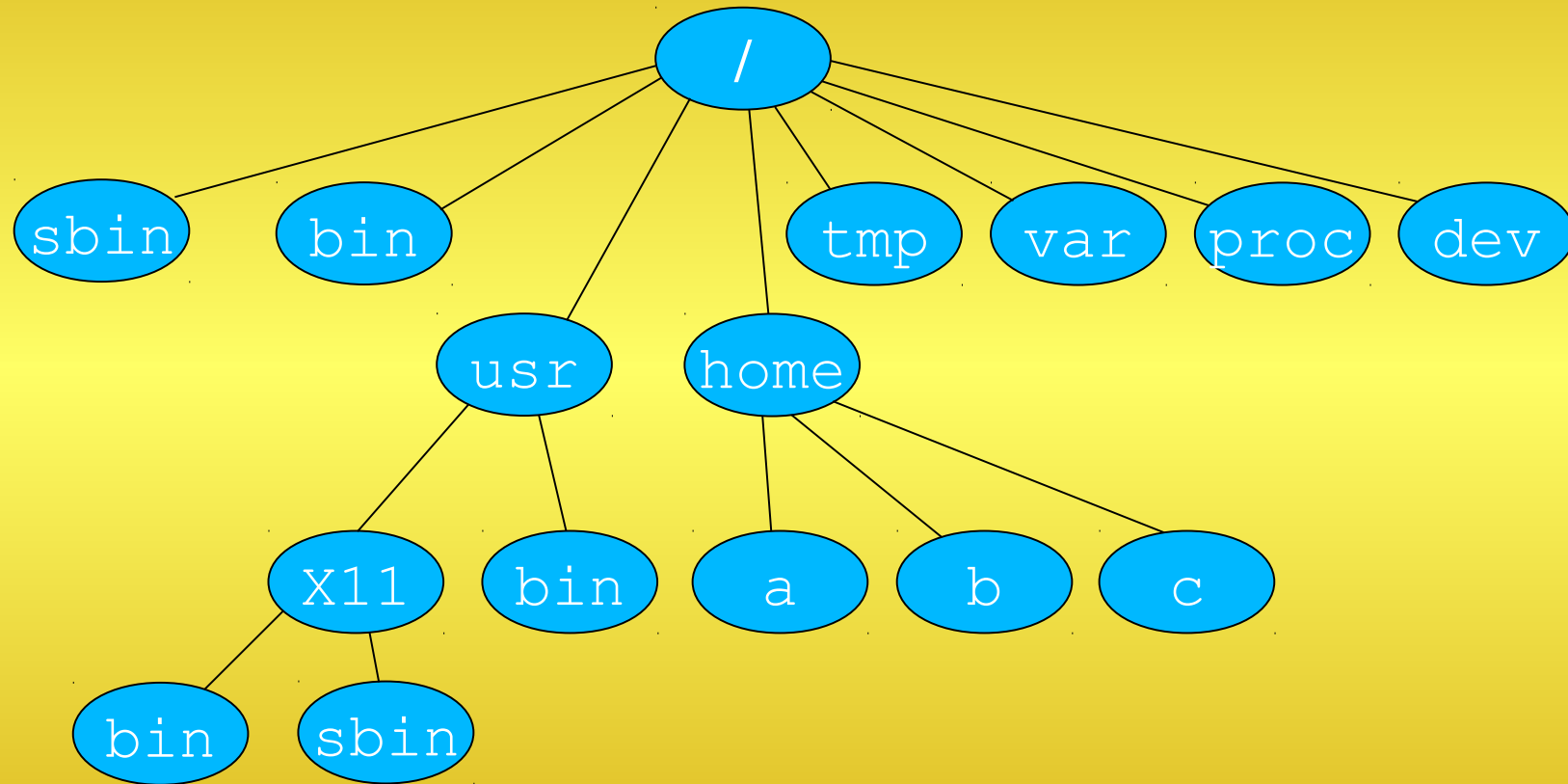
Aufbau eines Linuxsystems

- Linux:
 - bezeichnet eigentlich nur den Kernel
 - Unix ähnliches Betriebssystem
 - entwickelt von Linus Torvalds
 - unterliegt der GPL-Lizenz (open source)
- Multitasking / Multi-User System
- verfügbar für sehr viele Prozessortypen
- spezialisierte Versionen für eingebettete Systeme, Echtzeitsysteme, etc.
- mit Open-Source-Tools (GNU) vervollständigt.

Aufbau eines Linuxsystems 2

- hierarchisch aufgebautes System, d.h.
- jeder Systembestandteil wird als Datei repräsentiert
- jede Komponente durch Pfad eindeutig identifiziert
- absoluter Pfad: kompletter Pfad ausgehend vom Wurzelverzeichnis, beginnt immer mit „/“
- Relativer Pfad: ausgehend vom aktuellen Verzeichnis
- . steht für aktuelles Verzeichnis,
.. für übergeordnetes Verzeichnis
- Dateitypen: reguläre Datei, Verzeichnis, block – u. zeichenorientierte Geräte, Sockets, named pipes

Dateisystem - Überblickschema



<ftp://tsx-11.mit.edu/pub/linux/docs/linux-standards/fsstnd/>

Befehlsübersicht

- Dateisystem:
cd, ls, mkdir, rmdir, ln, touch, mv, pwd, rm, cp, chmod, chown, chgrp
- Dateien:
cat, less, head, tail, wc, strings, cut, paste, join, uniq, file
- Prozesse
ps, top, kill, killall
- Suche
grep, find, apropos, locate

Befehlsübersicht 2

- Archivierung
(g)zip, tar
- Systemstatus
df, du, env, date, who, stty, mount
- Dokumentation
man, info
- Benutzerverwaltung
passwd, id

Anmeldung am System

- Anmeldung am System auch „Login“ genannt:
 - Eingabe des Login-Namens (vom Systemadministrator vergeben)
 - Eingabe des Passwortes (unsichtbar)
- Änderung des Passwortes mit Hilfe des Kommandos **passwd**:
passwd *Username*
altes Passwort
neues Passwort (2x)

Informationen über Benutzer

- **id** *Loginname*
- liefert Informationen über User-Name, User-Id, die primäre und weitere Gruppen-Ids und Gruppennamen
- aktuelle Informationen über angemeldete Benutzer liefert das Kommando **who**

Online Dokumentation

- **man:** manual
 - ursprüngliches Unixhilfesystem
 - Manpages in 9 Sektionen eingeteilt
 - in eigener Datenbank indiziert
 - Eingrenzung der Suche auf Sektionen oder mit Schlüsselwörtern möglich
- **apropos:** Schlüsselsuche im Index, sucht alle Vorkommen in Dokumententiteln heraus

Online Dokumentation II

- **info:** GNU-Dokumentationswerkzeug
 - eigenes Hypertextsystem
 - Bedienung gewöhnungsbedürftig, aber sehr mächtig
 - in eigener Datenbank indiziert
 - eigentliche Informationsquelle für große Softwarepakete von GNU wie z.B GNU-Compiler, Latex, etc

Dateisystem 1

- Feststellen des aktuellsten Verzeichnisses

pwd: print working directory

- Wechseln in anderes Verzeichnis

cd: change directory

cd wechselt in Home-Verzeichnis

cd . . wechselt ein Verzeichnis Richtung
 Wurzel

cd *Pfad_zu_Ziel* wechselt ins Verzeichnis Ziel

Dateisystem 2

- Auflisten eines Verzeichnisses – Inhalt:
ls: list directory
- Wichtige Optionen:
 - a alle Dateien inklusive unsichtbare
 - l ausführliche Informationen zu den Dateien
 - d kein Inhalt von Verzeichnissen
 - t sortiert nach letztem Modifikationsdatum
 - r umgekehrte Sortierreihenfolge
 - h „human readable“, Größenangabe in Kilo-Megabyte
 - S sortiert nach Dateigröße

Dateisystem 3

- Erzeugen eines Verzeichnisses:
mkdir: make directory
mkdir *Verzeichnisname*
- Verzeichnis löschen:
rmdir: remove directory
rmdir *Verzeichnisname*
funktioniert nur bei leerem Verzeichnis

Dateisystem 4

- Kopieren von Dateien:
cp: copy files
cp *vonDatei nachDatei*
cp *vonDatei nachVerzeichnis*
- Optionen:
 - p erhält alle Informationen, Rechte, etc.
 - r -R rekursiv, zum Kopieren von Verzeichnissen

Dateisystem 5

- Verschieben bzw. Umbenennen von Dateien:

mv: move files

mv *vonDatei nachDatei*

mv *vonDatei nachVerzeichnis*

mv *vonVerzeichnis nachVerzeichnis*

- *Wichtig:*
 - vorhandene Zielformateien werden überschrieben
 - vorhandene Zielverzeichnisse nehmen verschobene Verzeichnisse auf

Dateisystem 6

- Löschen von Dateien:
rm: remove files
rm *Datei*
rm *-r Verzeichnis*
- *Optionen*:
 - r rekursives Löschen von Verzeichnissen
 - f unterdrückt Nachfragen beim Löschen schreibgeschützter Dateien
 - i interaktiv, Benutzer bestätigt Löschvorgang
- **WARNUNG**: Löschen ist unwiderruflich,
Anwendung von **rm -rf** nur mit großer Vorsicht

Dateisystem 7

- Erzeugen von Dateiverweisen:
ln: link files (Alias, Zweitname)
ln (*Pfad/*)*Datei Pfad/(LinkName)*
- *Optionen*:
 - s erzeugt einen symbolischen Link, d.h. eine eigene Verweisdatei
 - f unterdrückt Nachfragen beim Überschreiben vorhandener Dateien
- symbolische Links sind sicherer gegenüber unbeabsichtigtes Löschen der Zielfeile

Dateien und Rechte

- Jeder Benutzer hat eine ID und ist mindestens Mitglied einer Benutzergruppe
- Es können Rechte für den Besitzer, die Gruppe und übrige Benutzer vergeben werden.
- Die drei Gruppen werden durch u (Besitzer, owner), g (Gruppe, group) und o (übrige, other) sowie a (jeder Benutzer, all) bezeichnet.

Dateien und Rechte 2

- Es können folgende Rechte vergeben werden:
r read Leserechte
w write Schreibrechte
x execute Ausführungsrechte
s (set user/group ID), t(sticky bit) werden hier nicht weiter behandelt, siehe Man-Page
- Setzen eines Rechtes mit “+”, entziehen mit “-”
genaues Setzen mit =
- Angabe durch Buchstaben oder Oktalzahlen

Dateien und Rechte 3

- Beispiele:
ls -l gibt folgende Rechte aus:
-rw-r--r-- d.h. der Besitzer hat Schreib- und Leserechte auf der Datei, der Rest nur Leserecht
oktal 644
- Setzen des Schreibrechtes für die Gruppe:
chmod g+w *Datei* bzw. **chmod** 664 *Datei*
- Setzen des Ausführungsrechtes für alle:
chmod a+x *Datei* bzw. **chmod** 775 *Datei*

Dateien und Rechte 4

- Ausführungsrecht für Verzeichnisse erlaubt wechseln in das Verzeichnis
- Leserecht muss für Wechsel nicht gegeben sein
- Löschen einer schreibgeschützten Datei bei Schreibrecht auf Verzeichnis möglich !
- Optionen:
 - R rekursives Setzen der Rechte im Dateisystembaum

Dateien und Rechte 5

- „Übereignen“ von Dateien:
chown: change owner
chown *user.group Dateiname*
chgrp *group Dateiname*
- Optionen:
 - R rekursives Übereignen im Dateisystembaum

Dateien

- Anzeigen von Textdateien:
cat: catenate
cat *Dateiname*
Datei wird in einem Stück auf die Standardausgabe ausgegeben
- Komfortables Lesen von Textdateien:
less *Dateiname*
Ermöglicht seitenweise vor- und zurückblättern sowie Suche nach bestimmten Begriffen
- Ermitteln des Formats einer unbekannten Datei:
file *Dateiname*

Bearbeitung von Dateien

- Extraktion von Strings aus Binärdateien:
strings *Dateiname*
- Ausgabe der ersten n Zeilen einer Datei:
head -n *Datei*
- Ausgabe der letzten n Zeilen einer Datei:
tail -n *Datei*
- Option -f: zeigt jeweils die letzten aktuellen Zeilen an, gut für Überwachung von Logfiles

Bearbeitung von Dateien II

- Zählen der Zeilen, Wörter und Zeichen:
wc *Dateiname* (*word count*)
- Sortieren der Wörter:
sort *Dateiname*
- Zerlegen in Spalten bzw Felder:
cut
- Vereinigung von Dateien über gemeinsame Schlüssel:
join
- spaltenweise Vereinigung von Dateien:
paste

Bearbeitung von Dateien III

- Entfernen mehrfach vorkommenden Zeilen:
uniq
- Erzeugen einer leeren Datei oder aktualisieren
des Zeitstempels:
touch Dateiname

Systemstatus

- Anzeigen der Festplattenbelegung:
df: disk free
gibt den Platz auf vorhandenen Dateisystemen aus
- Platzverbrauch von Dateien und Verzeichnissen:
du: disk usage
gibt den Platzverbrauch von Dateien aus, viel
Optionen zur Verbesserung der Lesbarkeit
- eingehängte Dateisysteme(Laufwerke):
mount

Systemstatus II

- Anzeigen der Umgebungsvariablen:
env: environment
gibt die bereits gesetzten Umgebungsvariablen der Shell aus
- Anzeige der Systemzeit:
date
- Zeitverbrauch eines Programms:
time
- eingeloggte Benutzer und Prozesse: **who**
- Terminaleinstellungen: **stty**
 - wenn das Terminal verstellt ist: **stty sane**

Prozesse

- Prozeß: Vorgang, der vom Betriebssystem Speicher und Rechenzeit erhält
- Charakterisiert durch: eigene ID, Elter-ID (parent process), verbrauchte Rechenzeit, belegter (virtueller) Speicher, Zustand
- kann im Vordergrund oder Hintergrund ablaufen
- kann auch ohne Steuerterminal laufen (Dämonprozeß)

Prozesse 2

- **ps**: report process status
ps [optionen]
- Optionen:
 - u UID Prozesse für Benutzer UID,
ohne Angabe von UID für alle Benutzer
 - a alle Prozesse mit einem Terminal
 - x zeigt Prozesse ohne Kontrollterminal
 - l ausführliche Informationen (long)
 - f zeigt Prozessbaum

Prozesse 3

- **kill**: kill processes
kill -Signal PID
- Optionen:
Signal 15 (terminate, geordnet beenden),
 9 (abort, sofort unbedingt beenden)
- **killall** ProgName
Beendet alle Prozesse von Programm ProgName,
vorausgesetzt, die Rechte erlauben es
- **top**: Tool zur Online-Anzeige aktiver Prozesse

Piping

- Viele Werkzeuge benutzen als Standardausgabe den Bildschirm und als Standardeingabe die Tastatur
- Diese Standardkanäle können umgelenkt und somit zu einer Kette von Verarbeitungsschritten verknüpft werden (piping)

Einfache Ein/Ausgabeumleitung

- Umleitung der Bildschirmausgabe in eine Datei durch folgende Ausführung:
`meinProgramm > meineAusgabedatei`
- Umleitung der Tastatureingabe aus einer Datei durch folgende Ausführung:
`meinProgramm < meineEingabedatei`
- Kombination beider Möglichkeiten:
`mProgramm > mAusgabedatei < mEingabedatei`

Piping

- Pipes stellen einen Betriebssystemmechanismus zur automatischen Verbindung hintereinandergeschalteter Programme dar:
Prog1 | Prog2 | Prog3
d.h. die Ausgabe von Programm 1 wird direkt als Eingabe an Programm 2 weitergereicht und die Ausgabe von Programm 2 an Programm 3
- Ziel: viele kleine spezialisierte Tools werden für komplexe Aufgaben kombiniert

Filtering

- Bei einer Kette hintereinandergeschalteter Programme findet in der Regel auf jeder Stufe eine Prozessierung statt. D.h. bestimmte Daten werden verändert oder herausgefiltert, daher wird dieser Vorgang auch „Filtering“ genannt.

grep

- **grep**: get regular expression
- Suche von Mustern in Eingabe:
 - Muster sind dabei reguläre Ausdrücke
 - Reguläre Ausdrücke werden ausführlich bei Perl behandelt
- Eingabe: Dateien oder Standareingabe
- Optionen:
 - c zählt Vorkommen von Mustern (count)
 - r durchsucht Verzeichnisse rekursiv
 - e spezifiziert Suchmuster
 - v alle Zeilen, die Muster nicht enthalten
 -

find

- **find** finde Dateien
- durchsucht ausgehend von einem Startpunkt das Dateisystem nach (bestimmten) Dateien
- die gewünschten Dateien werden durch Bedingungen spezifiziert
- Bedingungen / Optionen:
 - name, -type, -size, -ctime, -mtime, -depth.
 - follow, etc.

locate / which

- **locate**: Auffinden von Dateien
 - benutzt eigene Datenbank
 - Datenbank wird von Cron-Job regelmäßig aktualisiert
 - unterstützt ebenfalls Suchmuster
 - schneller als **find**, aber abhängig von Datenbank
- **which**: Auffinden ausführbarer Programme im Suchpfad, Programme werden gefunden falls sie in einem Verzeichnis in der PATH-Variable stehen

Archivierung

- **(g) zip**: Komprimieren von Dateien
(g) unzip: entsprechend zum Entpacken
- **tar** : tape archive
 - kombinierbar mit g(un)zip zur Kompression des Archivs
 - wichtigste Optionen: c,x,t
 - Bsp: **tar -czf archiv.tgz verzeichnis/** zum Einpacken von Verzeichnis in die Datei *archiv.tgz*
 - **tar -xzf archiv.tgz** zum Entpacken des Archivs *archiv.tgz* ins Verzeichnis *Verzeichnis*

Beispiele Piping

- `find /usr/local/include -name "*.h" | grep gnu *|wc`
- `cd ; ls .* | wc -l`
- `cat /etc/passwd/ | grep system | less`
- `cd; find -type f | grep -c mp3`

Weitere Tools

- **tee** *Dateiname*: liest Standardeingabe und schreibt sowohl nach Standardausgabe als auch in Datei
- **sort**: Eingabezeilen werden je nach Option numerisch oder alphabetisch sortiert

Mehr Beispiele

- `cat /etc/passwd | cut -f1,5 -d:`
- `find . -name *.pl | wc`
- `cd; du -h --max-depth=3 | sort -rn
| less`
- `cd; grep -lvir gnu * | wc -l`